

JARINGAN SARAF TIRUAN SEBAGAI ALTERNATIF UNTUK PENYELESAIAN TRAVELLING SALESPERSON PROBLEM

Kartika Gunadi

Fakultas Teknologi Industri, Jurusan Teknik Informatika - Universitas Kristen Petra
e-mail: kgunadi@petra.ac.id

Peter Iksan

Alumnus Fakultas Teknologi Industri, Jurusan Teknik Elektro - Universitas Kristen Petra

ABSTRAK: Traveling Salesperson Problem (TSP) adalah problem optimasi kombinasional yang tergolong dalam NP-complete problem. TSP adalah problem untuk menentukan urutan dari sejumlah kota yang harus dilalui oleh seorang sales, setiap kota hanya boleh dilalui sekali dan hanya sekali dalam perjalanan, dan perjalanan berakhir pada kota awal dimana seorang sales memulai perjalanannya. TSP ini dapat dilakukan secara sederhana dengan Algoritma Exhaustive, yaitu dengan mencari semua kombinasi yang mungkin terjadi, kemudian memilih kombinasi dengan jarak terdekat. Algoritma Exhaustive ini menjadi tidak efisien bila jumlah kota yang besar, karena mempunyai kompleksitas sebesar $n!/2n$. Jaringan saraf tiruan (JST) dapat digunakan untuk menyelesaikan problem optimasi dengan memilih arsitektur jaringan yang sesuai untuk mendapatkan solusi yang optimal. Algoritma dengan menggunakan jaringan saraf tiruan memberikan reduksi waktu eksekusi yang sangat signifikan untuk jumlah kota lebih besar 9, dan dapat memberikan persentase optimasi sebesar 83 % dari solusi yang terbaik yang didapatkan dengan algoritma exhaustive.

Kata kunci: Traveling Salesperson Problem (TSP), Jaringan saraf tiruan (JST), Algoritma Exhaustive, JST Hopfield

ABSTRACT : *Traveling Salesperson Problem (TSP) is one among the NP-complete problem. TSP is defined as follows. A traveling salesperson has a number of cities to visit. The sequence in which the salesperson visits different cities is called a tour. A tour should be such that every city is visited once and only once. The goal is to find a tour that minimize the total distance of a tour. TSP can be solved by Exhaustive algorithm, but this algorithm is no efficient for large number of cities. A neural network can be used to solve this optimization problem by choosing appropriate architecture to find optimal solution. A Neural network can be used to solve optimization by choosing appropriate architecture to find optimal solution. Algorithm of Neural network reduced a significant execution time for number of cities more than 9, and has optimization 83% of best solution from exhaustive algorithm.*

Keywords: Traveling Salesperson Problem (TSP), Neural network, Exhaustive algorithm, Hopfield neural network.

1. PENDAHULUAN

Traveling Salesperson Problem (TSP) adalah problem optimasi yang tergolong dalam NP-complete problem. TSP adalah problem untuk menentukan urutan dari sejumlah kota yang harus dilalui oleh seorang sales, setiap kota hanya boleh dilalui sekali dan hanya sekali dalam perjalanan, dan perjalanan berakhir pada kota awal dimana seorang sales memulai perjalanan.

Jumlah kombinasi rute perjalanan untuk sejumlah n kota sebanyak $n!/2n$. Sebagai contoh untuk $n=10$ maka didapat 181.400

kombinasi, untuk $n=15$ didapat kombinasi lebih dari 43 milyar, dan untuk $n=17$ maka didapat kombinasi lebih dari 1 triliun. Hal ini menunjukkan bahwa Algoritma Exhaustive ini menjadi tidak efisien bila jumlah kota lebih dari 10 kota.

2. MODEL MATEMATIK TSP

TSP dapat dinyatakan dalam model matematik di bawah ini:

$$\text{Minimasi} = \sum_{i=1}^n \sum_{j=1}^n X_{ij} \times D_{ij} \quad (1)$$

Dengan batasan :

$$\sum_{i=1, j \neq i}^n X_{ij} = 1, \text{ untuk } j=1,2,\dots,N \quad (2)$$

$$\sum_{j=1, i \neq j}^n X_{ij} = 1, \text{ untuk } j=1,2,\dots,N \quad (3)$$

$$U_i - U_j + N_{xij} \leq N - 1 \text{ untuk } i \neq j, i=2,3,\dots,N, j=2,3,\dots,N \quad (4)$$

$$U_i > 0, U_i \text{ adalah urutan kota ke } j \text{ dalam tour} \quad (5)$$

$$X_{ij} = 0 \text{ atau } 1 \text{ untuk semua } i \text{ dan } j \quad (6)$$

3. ALGORITHMHA EXHAUSTIVE

Algorithmha untuk menyelesaikan TSP ini dapat dilakukan secara sederhana dengan Algorithmha Exhaustive, yaitu dengan mencari semua kombinasi yang mungkin terjadi, kemudian memilih kombinasi perjalanan dengan jarak terdekat, algorithmha ini mempunyai kompleksitas $n!/2n$.

Sebagai contoh untuk 4 kota, maka didapatkan kombinasi rute perjalanan sebanyak 3 ($4!/2/4$) kombinasi.

Tabel.1 Jarak antar kota

KOTA	A	B	C	D
A	0	10	14	7
B	10	0	6	12
C	14	6	0	9
D	7	12	9	0

Kombinasi rute perjalananya adalah:

1. A B C D A = 32
2. A C D B A = 45
3. A D B C A = 39

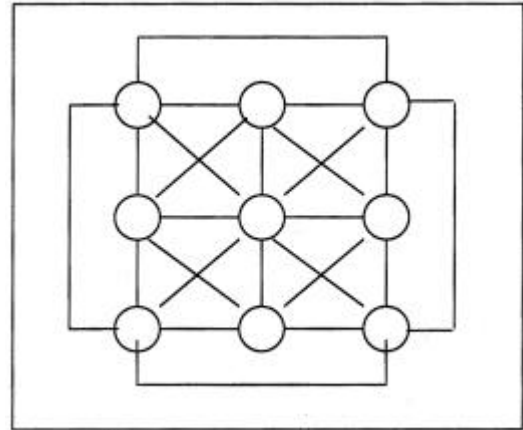
Rute perjalanan yang optimum adalah: A B C D A sejauh = 32

4. JARINGAN SARAF TIRUAN

Jaringan saraf tiruan adalah sistem pengolahan informasi yang mempunyai sifat karakteristik yang menyerupai sifat karakteristik jaringan saraf biologis. Sifat karakteristik ini meliputi pola koneksi antar neuron, metoda untuk menentukan bobot pada koneksi pada neuron dan fungsi aktivasi pada neuron.

Jaringan saraf tiruan Hopfield terdiri dari n^2 neuron yang disusun dalam larik 2

dimensi, dimana jumlah baris sebanyak n dan jumlah kolom sebanyak n . Antar neuron terdapat koneksi penuh dengan semua neuron yang lain. Koneksi neuron dalam jaringan untuk setiap baris dan setiap kolom terkoneksi secara lateral.



Gambar 1. JST Hopfield

Algorithmha JST Hopfield

- Mulai
- Input jarak antar kota
- Unisialisasi output lintasan secara acak
- Iterasi sampai kondisi dibawah ini terpenuhi:
 - Pada setiap baris hanya ada sebuah "1"
 - Pada setiap kolom hanya ada sebuah "1"
 - Jumlah "1" sebanyak n (jumlah kota) didapatkan energi minimum
- Hitung jarak rute perjalanan
- Tampilkan lintasan dan jarak rute perjalanan
- Selesai

5. IMPLEMENTASI

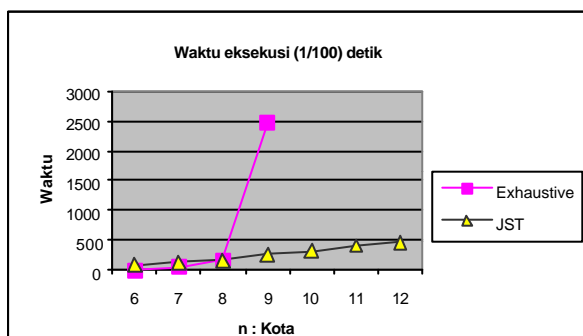
Untuk mendapatkan hasil kerja dari suatu algorithmha, maka algorithmha optimasi TSP perlu diimplementasikan kedalam bahasa pemrograman dan dilakukan pengukuran waktu komputasi. Pada penulisan ini algorithmha optimasi diimplementasikan kedalam bahasa pemrograman Pascal, dengan menggunakan compiler Turbo Pascal. Program dijalankan pada komputer personal dengan prosesor Pentium 233 mmx dengan memory 64 MB.

6. HASIL PENGUKURAN

Pengukuran waktu eksekusi dilakukan dengan berbagai jumlah kota, karena sifat dari algoritma pada JST yang menggunakan bilangan acak, maka pengukuran dilakukan sebanyak 10 kali untuk setiap jumlah kota kemudian dilakukan perhitungan rata-rata, hal ini dilakukan dengan tujuan untuk mendapatkan ketelitian dari pengukuran. Satuan pengukuran terkecil adalah sebesar 1/100 detik, satuan yang mampu diberikan oleh bahasa pemrograman Pascal.

Tabel 2. Waktu Eksekusi (1/100 detik)

Jumlah Kota	Algoritma Exhaustive (3)	Jst Hopfield
6	0	88
7	50	130
8	170	165
9	2470	270
10	25480	314
11	280280	410
12	3363360	470



Gambar 2. Waktu eksekusi Algoritma Exhaustive vs JST

Persentase optimasi rute perjalanan yang didapat pada JST didefinisikan sebagai rasio terhadap nilai optimum yang didapat pada Algoritma Exhaustive.

$$\text{Optimasi} = 1 - \frac{(\text{SolusiJST} - \text{Solusiterbaik Algoritma Exhaustive})}{(\text{Solusiterbaik Algoritma Exhaustive} - \text{Solusiterjelek Algoritma Exhaustive})} \times 100$$

Tabel 3. Persentase Optimasi (%)

Jumlah Kota	JST Hopfield (%)
6	80,01
7	83,10
8	81,65
9	82,70
10	82,97
11	84,14
12	84,38

7. KESIMPULAN

Lama waktu eksekusi Algoritma Exhaustive efisien untuk menyelesaikan problem TSP dengan jumlah kota sampai dengan 10 buah, dan mulai menunjukkan peningkatan yang sangat drastis pada jumlah kota sebanyak 11 buah. Sedangkan JST Hopfield secara umum peningkatan waktu eksekusi tidak menunjukkan peningkatan yang drastis.

Solusi optimal selalu dapat didapatkan pada Algoritma Exhaustive karena sifatnya yang mengevaluasi dari semua kombinasi yang ada, sedangkan JST Hopfield memberikan persentase optimasi sebesar 83 %.

DAFTAR PUSTAKA

1. Valuru B. Rao, Hayagriva V. Rao, *C++ Neural Networks and Fuzzy Logic*, MIS Press, 1993.
2. Laurene Fausett, *Fundamentals of Neural Network*, Prentice Hall, 1994.
3. *TSP*, Knowledge Garden. Inc., 1997.
4. Peter Iksan, *Studi Banding Algoritma Konvensional dengan Algoritma Jaringan Saraf Tiruan Pada Traveling Salesperson Problem*, Tugas akhir, 1999.